

IN THE CLAIMS:

Please amend the claims as follows. The claims are in the format as required by 35 C.F.R. § 1.121.

1. (Currently amended) A method of modeling an arbitrarily complex environment, comprising:

on a computer having a computer memory and a processor, defining, on a model computer, a schema having a plurality of types of data structures in a data model, wherein each of the data structure[[s]] comprises one or more fields or properties associated with the data structure, wherein all data structures of the same type contain the same properties;

storing, by a database computer, the schema in a static database, wherein the static database comprises a table for each type of data structure, wherein the plurality of types of data structures comprises components, relationships, properties and types

instantiating a component for each atomic entity in the arbitrarily complex environment wherein each component has a set of fields which contain information relating to the atomic entity associated with the component, wherein the set of fields comprises:

a set of property fields containing information about the attributes or characteristics of the component; and

a field that contains a link to its component type; and
representing, by a database connectivity computer, entities in the arbitrarily complex environment by assigning one or more values to the properties one or more fields in the instantiated component database based on the attributes of the entity which the component was instantiated to represent;

instantiating a relationship for representing an association or a dependency between two or more components in the data model, wherein each relationship comprises:

a field that is a foreign key to its relationship type; and

a set of property fields containing information about one or more of the attributes of the relationship; and

storing the components in a schema, wherein property definitions of each component are linked to a type of component,

wherein adding to or altering the data model further comprises the steps of:
accessing, by a manager computer, the model computer, to add or alter a data structure; and

storing, by the model computer, the added or altered data structure to the database, wherein changes made to the type of component are automatically associated with all components of that type of component without changing the schema to reflect a corresponding change in the arbitrarily complex environment wherein the addition or alteration of components, relationships, properties and types do not change the defined data structures or the schema.

2. (Currently amended) The method of claim 1, wherein each component is instantiated based on a generic component type and has a set of core attributes comprising an id, a name, a description, a type, and a set of properties and the schema comprises:

a propertyDefinition table, a componentType table, a component table, a propertyTable category, a propertyDefinition table, a propertyValue table, a relationshipType table and a relationship table, wherein each table comprises a set or properties.

3. (Previously Presented) The method of claim 1, wherein each component type is in a hierarchy of component types.

4. (Currently amended) The method of claim 2, wherein [[each]] a property [[has]] comprises a data type of one of a string, a numeric, a Boolean, a link, a date/time and a custom type.

5. (Currently amended) The method of claim 2, wherein each property [[is]] comprises a data structure having a name, a description and a value.

6-9. (Cancelled).

10. (Currently amended) The method of claim [[1]] 2, wherein each component is represented in [[a]] the component table, wherein each component stored in the component table is linked to a type of component.

11. (Currently amended) The method of claim 10, wherein each component type is represented in the component type table, wherein each component type stored in the component type table is linked to a property in the property table.

12-15. (Canceled).

16. (Currently amended) A system for modeling an arbitrarily complex environment, comprising:

a computer having a memory for storing a set of computer-executable instructions and a processor for executing the computer-executable instructions operable to:

define a schema having a plurality of data structures in a data model, wherein each of the data structures comprises one or more fields of properties associated with the data structure, wherein all data structures of the same type contain the same properties;

instantiate a set of components in the data model, wherein each component in the set of components represents a particular atomic entity within the arbitrarily complex environment, wherein each component has a set of fields which contain information relating to the atomic entity associated with each component, wherein the set of fields comprises;

a property field containing information about the attributes or characteristics of the component; and

represent entities in the arbitrarily complex environment by assigning one or more values to the properties fields in the instantiated component database based on the attributes of the entity which the component was instantiated to represent;

instantiate a set of relationships in the data model, wherein each relationship represents an association between at least two of the components, wherein each relationship comprises:

a property field containing information about one or more of the attributes of the relationship; and

wherein, when the data model is added to or altered, the system is further operable to:

store the added or altered data structure to the database components in a schema, wherein property definitions of each component are linked to a type of component, wherein changes made to the type of component are automatically associated with all components of that type of component without changing the schema, and wherein one or more fields in a relationship are changed based on a check in the relationship check field to reflect a corresponding change in the arbitrarily complex environment, wherein the addition or alteration of components, relationships, properties and types do not change the defined data structures or the schema.

17. (Currently amended) The system of claim 16, wherein each component is instantiated based on a generic component type and has a set of core attributes comprising an id, a name, a description, a type, and a set of properties and the schema comprises:

a propertyDefinition table, a componentType table, a component table, a propertyTable category, a propertyDefinition table, a propertyValue table, a relationshipType table and a relationship table, wherein each table comprises a set or properties.

18. (Currently amended) The system of claim ~~[[16]]~~ 17, wherein each component type is in a hierarchy of component types.

19. (Currently amended) The system of claim 18, wherein ~~[[each]]~~ a property ~~[[has]]~~ comprises a data type of one of a string, a numeric, a Boolean, a link, a date/time and a custom type.

20. (Currently amended) The system of claim ~~[[19]]~~ 18, wherein ~~[[each]]~~ a property ~~[[is]]~~ comprises a data structure having a name, a description and a value.

21-22. (Cancelled).

23. (Currently amended) The system of claim ~~[[16]]~~ 17, wherein each relationship type is a parent type or a subtype.

24-30. (Canceled).

31. (Currently amended) A software product comprising a set of instructions stored on a computer-readable medium, wherein the computer has a computer memory and a processor for executing the set of instructions, wherein the software product comprises:

an instruction to define a schema having a plurality of data structures in a data model, wherein each of the data structures comprises one or more fields ~~of properties associated with the data structure, wherein all data structures of the same type contain the same properties;~~

~~an instruction to instantiate a component for each atomic entity in the arbitrarily complex environment, wherein each component has a set of fields which contain information relating to the atomic entity associated with the component, wherein the set of fields comprises:~~

~~a property field containing information about the attributes or characteristics of the component; and~~

~~an instruction to represent entities in the arbitrarily complex environment by assigning values to the properties fields in the instantiated component database based on the attributes of the entity which the component was instantiated to represent;~~

~~an instruction to instantiate a relationship to represent an association or a dependency between two or more components, wherein each relationship comprises:~~

~~a property field containing information about one or more of the attributes of the relationship; and~~

~~wherein, when the data model is added to or altered, the software product further comprises an instruction to~~

~~store the added or altered data structure to the database components in a schema, wherein property definitions of each component are linked to a type of component, wherein changes made to the type of component are automatically associated with all components of that type of component without changing the schema, wherein one or more fields in a relationship are changed to reflect a corresponding change in the arbitrarily complex environment, wherein the addition or alteration of components, relationships, properties and types do not change the defined data structures or the schema.~~

32. (Currently amended) The software product of claim 31, wherein each component is instantiated based on a generic component type and has a set of core attributes comprising an id, a name, a description, a type, and a set of properties and the schema comprises:

a propertyDefinition table, a componentType table, a component table, a propertyTable category, a propertyDefinition table, a propertyValue table, a relationshipType table and a relationship table, wherein each table comprises a set or properties.

33. (Currently amended) The software product of claim ~~[[31]]~~ 32, wherein each component type is in a hierarchy of component types.

34. (Currently amended) The software product of claim 33, wherein ~~[[each]]~~ a property ~~[[has]]~~ comprises a data type of one of a string, a numeric, a Boolean, a link, a date/time and a custom type.

35. (Currently amended) The software product of claim ~~[[34]]~~ 33, wherein ~~[[each]]~~ a property ~~[[is]]~~ comprises a data structure having a name, a description and a value.

36-37. (Cancelled).

38. (Previously Presented) The software product of claim 37, wherein each relationship type is a parent type or a subtype.

39-53. (Cancelled).